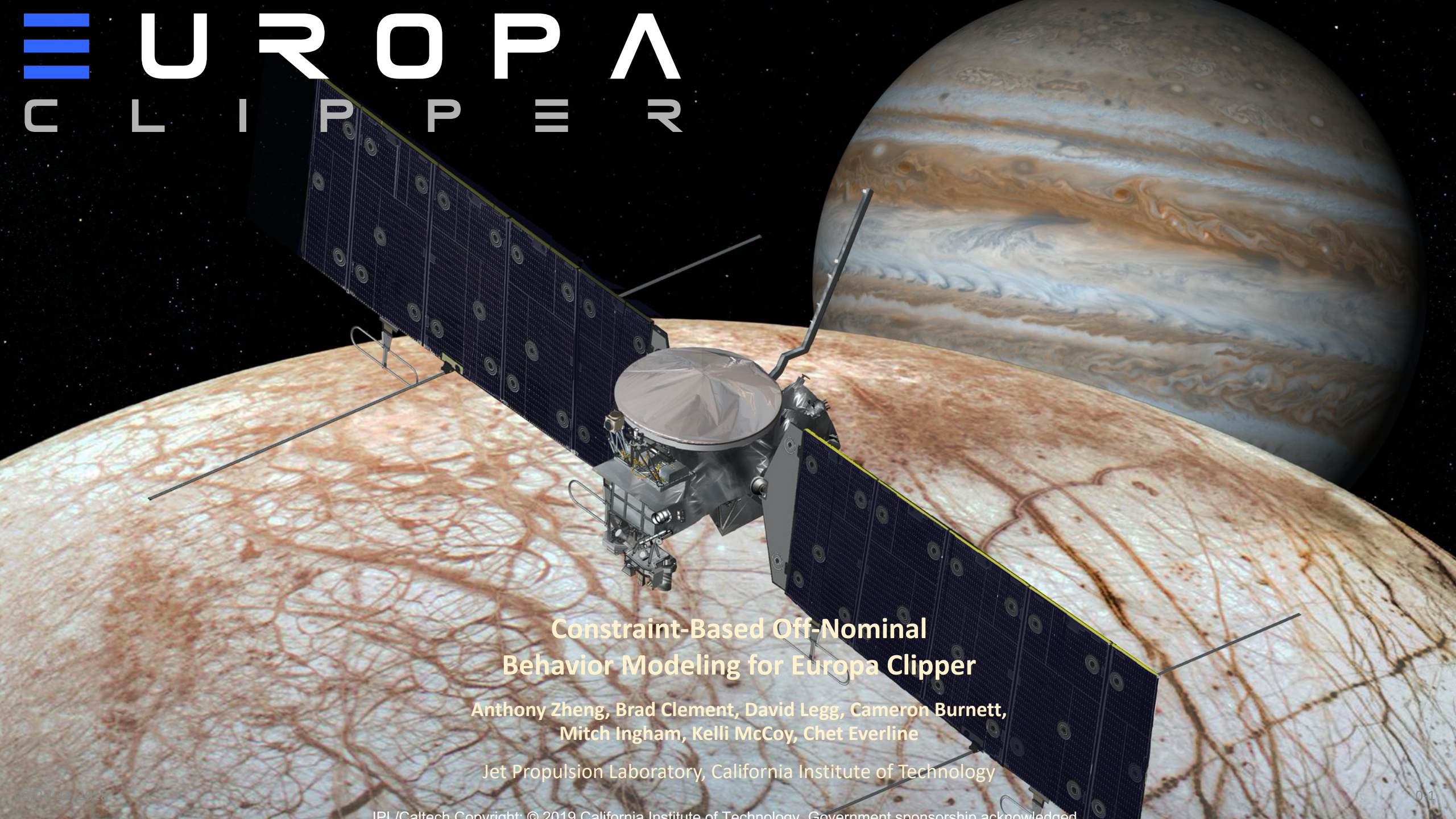


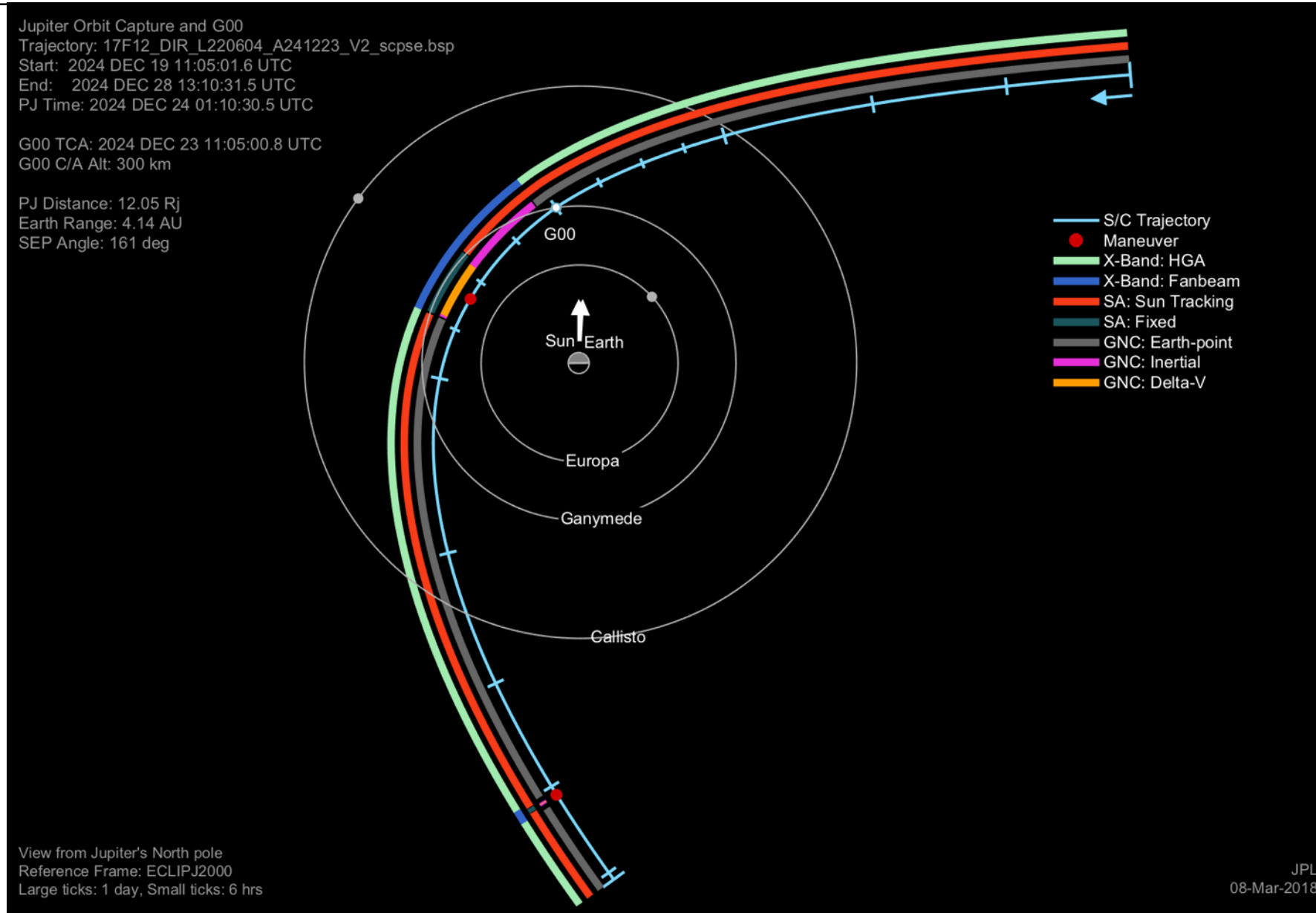
Constraint-Based Off-Nominal Behavior Modeling for Europa Clipper

Anthony Zheng, Brad Clement, David Legg, Cameron Burnett, Mitch Ingham, Kelli McCoy, Chet Everline

Jet Propulsion Laboratory, California Institute of Technology

Overview: Motivation

- Jupiter orbit insertion (JOI) is a project critical event, which if missed, could mean loss of mission.
- A probabilistic risk assessment (PRA) is being performed so the project can understand its risk posture for the JOI event.



Background

- Existing software to compute probability of spacecraft/mission failure supports models of
  - dependencies among elements of spacecraft
    - component Z fails if either X or Y fails
  - vulnerability to faults when powered/unpowered
  - potential recovery from faults
- Assumes a fixed execution schedule with minor deviations
- See

  S. Schreiner, M. L. Rozek, A. Kurum, C. J. Everline, M. D. Ingham, and J. Nunes, “Towards a methodology and tooling for Model-Based Probabilistic Risk Assessment (PRA),” in AIAA SPACE 2016.

Problem: Limited ability to represent fault responses

- Existing PRA modeling does not allow for
  - faults that depend on other faults
    - Example: a failure to swap from component A to redundant component B only happens if there is a fault to A to begin with.
  - changes in activities in response to a fault response
    - Example: if the battery is too low during orbit insertion, propulsion is interrupted to recharge the batteries.
- Analytic computations for special cases can be challenging to both formulate and compute.

JOI Thruster Faults

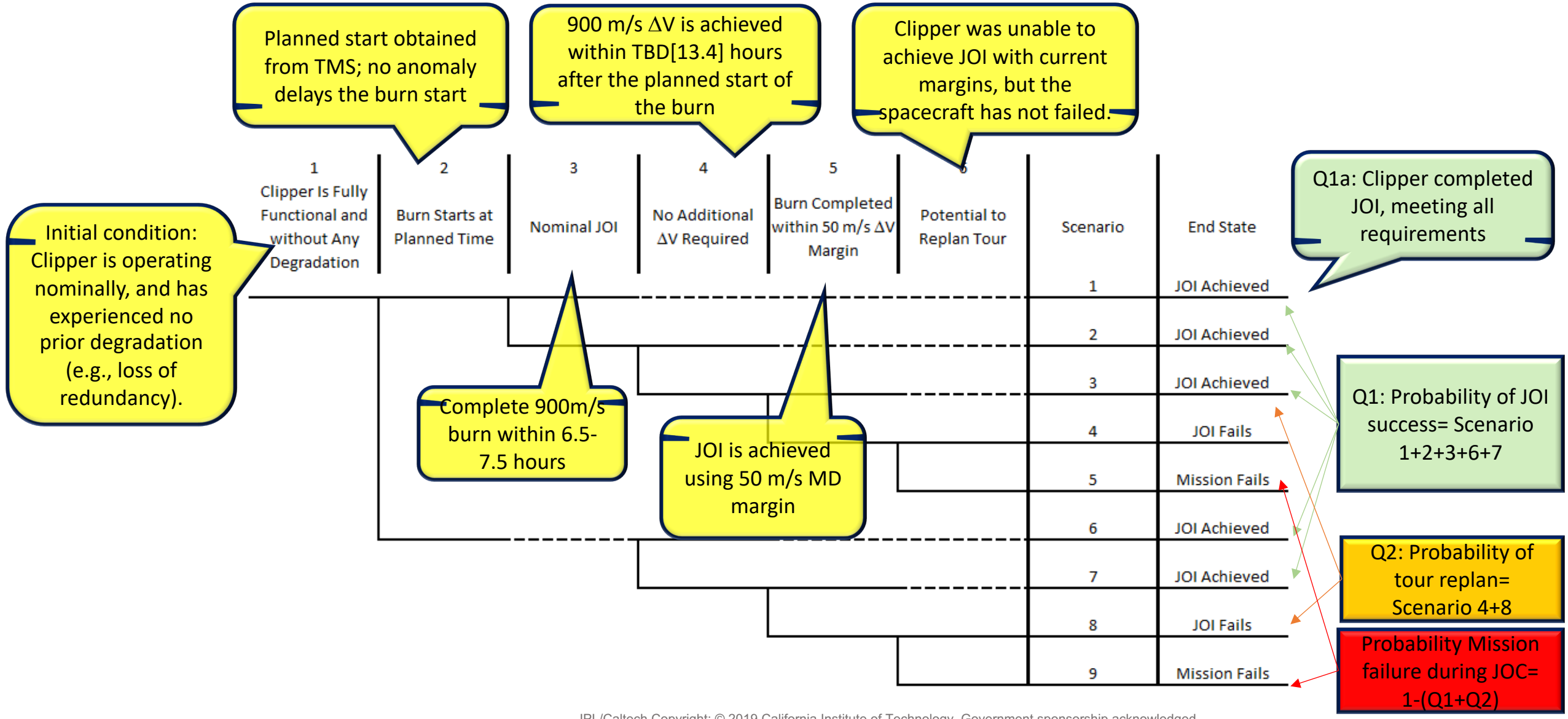
- Eight thrusters fire during nominal JOI for 6.5 hours.
- A fault in thrusters will swap a pair (for balance) to backup thrusters.
- If the backup fails, then the thrusters are reduced to six.
- If more thrusters & backups fail, four thrusters are used.
- A total of 900 m/s of delta-v (change in velocity) must be achieved for JOI success.
- If there are fewer thrusters, it takes longer to achieve the delta-v goal.
- If it takes too long, additional delta-v must be added on top of 900 m/s.
- Probability of success is calculated by integrating a reliability formula over time:

$$\int_0^{6.5} \int_x^{S_1(6.5-x)} \int_y^{S_2\left(6.5-x-\frac{1}{S_1}y\right)} r(x, y, z, T)\, dz\, dy\, dx$$

  x, y, and z are durations of thrusting with 8, 6, and 4 thrusters, respectively.

- For different scenarios, we need to determine these possible duration values to know what bounds to use for the integration.
- Note: the actual handling of thruster faults may be different for Europa Clipper.

JOI Event Tree: Scenarios 1 - 9



Determining Risk Probabilities

- To calculate the probability of these scenarios occurring, we need to determine the possible durations on 8, 6, and 4 thrusters for each scenario.
- We represent this as an optimization problem.
- The different scenarios are have different constraints on completion time and additional delta-v.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
JOI Completion Time (hours)	$6.5 \leq t \leq 7.5$	$7.5 \leq t \leq 13.4$	$t > 13.4$	$t > 13.4$
Additional Δv (m/s)	$\Delta v = 0$	$\Delta v = 0$	$0 < \Delta v < 50$	$\Delta v > 50$

Finding Time Bounds

- We frame finding time bounds as an optimization problem – minimize or maximize the time spent on a particular number of thrusters before a fault.
- Tried four approaches:

Approach 1: Solve by hand.

- Able to do scenarios 1 to 4 with some difficulty. Other scenarios too difficult.

Approach 2: Use a mixed integer linear program (MILP) solver.

- Difficult to encode the problem in languages such as AMPL.
- The problem turned out to be too difficult for easily accessible toolkits.

Approach 3: Behavior Analysis Engine (BAE) https://github.com/Open-MBEE/kservices

- Easy to encode the problem in the K language.
- BAE can solve for point solutions, but we need the bounds expressed as formulas.

Approach 4: Use Mathematica to solve

- Mathematica is able to solve and simplify to get the bounds as formulas.
- Special knowledge to configure Mathematica to solve efficiently and present solution.

Results for a single scenario

- 7 cases of different mixes of numbers of thrusters.

Case	Constraints	Δt	Δt_8	Δt_6	Δt_4
8 thrusters throughout	$\Delta t_8 > 0$ $\Delta t_6 = 0$ $\Delta t_4 = 0$	Not possible			
Switch from 8 to 6 thrusters	$\Delta t_8 > 0$ $\Delta t_6 > 0$ $\Delta t_4 = 0$	$7.5 < \Delta t < \frac{4}{3} \cdot 6.5$	$0.62 < \Delta t_8 < 3.5$	$\frac{4}{3}(6.5 - \Delta t_8)$	0

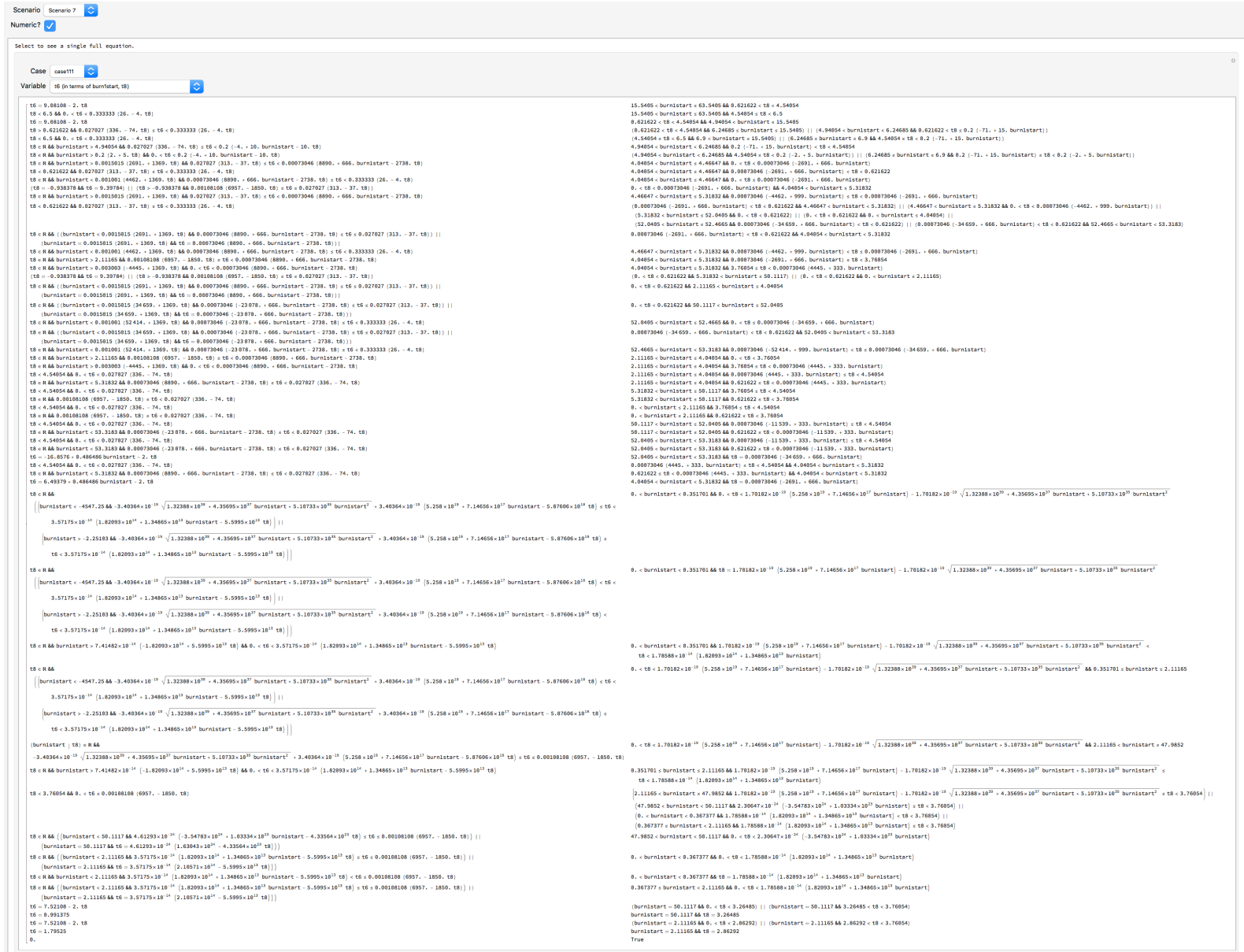
Switch from 8 to 6 to 4 thrusters	$\Delta t_8 > 0$
	$\Delta t_6 > 0$
	$\Delta t_4 > 0$
		$7.5 < \Delta t < 2 \cdot 6.5$	$0.62 < \Delta t_8 < 5.5$	if $\Delta t_8 < 4.54$ then
				$4.54 \cdot 2 - 2\Delta t_8 \leq \Delta t_6 < \frac{4}{3}(6.5 - \Delta t_8)$
				if $\Delta t_8 \geq 4.54$ then
				$0 < \Delta t_6 < \frac{4}{3}(6.5 - \Delta t_8)$
					$2\left(6.5 - \Delta t_8 - \frac{3}{4}\Delta t_6\right)$

Switch from 8 to 4 thrusters	$\Delta t_8 > 0$ $\Delta t_6 = 0$ $\Delta t_4 > 0$	$7.5 < \Delta t < 2 \cdot 6.5$	$4.54 < \Delta t_8 < 5.5$	0	$2(6.5 - \Delta t_8)$
6 thrusters throughout	$\Delta t_8 = 0$ $\Delta t_6 > 0$ $\Delta t_4 = 0$	Not possible			
Switch from 6 to 4 thrusters	$\Delta t_8 = 0$ $\Delta t_6 > 0$ $\Delta t_4 > 0$	Not possible			
4 thrusters throughout	$\Delta t_8 = 0$ $\Delta t_6 = 0$ $\Delta t_4 > 0$	Not possible			

Additional challenges

- It was difficult to get Mathematica to solve the problem.
  - Mathematica would often run out of memory and crash.
  - Other times it would compute for hours before we aborted.
  - We eventually found a way to break up the problem into many smaller problems.
  - Using 7 threads on a 2018 Macbook Pro, it takes 20 minutes.
- The solutions to scenarios 6, 7, and 8 have very large formulas.
  - There may be a way to simplify them, but it's not obvious and Mathematica can't do it.
- If the problem were any bigger, we don't think we could get an analytic solution.
- The alternative is an empirical approach, like Monte Carlo.
- Analytic formulations are best because they are faster to recompute for changes to inputs.
- But formulating an analytic model is often difficult and time-consuming.
- Monte Carlo takes longer to converge to an answer, but formulating the problem is much easier.
- Mixing analytic and empirical formulations may be most efficient.

Screenshot of formula for t6 in terms of t8 and burn1start for Scenario 7, t8>0, t6>0, t4>0



Summary

- The pre-existing PRA software does not handle
  - faults that depend on other faults and
  - fault responses that change the activities scheduled for execution
- Fault chains can result in nested integrals over time for calculating risk analytically.
- Determining the time bounds for these integrals for degraded thrusting is challenging.
- Ultimately, Mathematica was the only tool that would provide formulas for the time bounds.
- While analytic formulations are better than empirical, they are not always feasible.
- Is there a general approach for mixing analytic and empirical?